

CLAIMS

What is claimed is:

- 5 1. A file system for a nonvolatile memory medium, comprising:
- a nonvolatile memory medium logically divided into memory blocks of
predetermined size, wherein each of said memory blocks comprises a header
portion and a data portion, and wherein said header portion comprises:
- a magic number indicating whether said memory block is a valid file
10 system block;
- the name of the file to which said memory block belongs, if any;
- a set of flags indicating whether said memory block is the first or last
block of said file;
- the block number of the next memory block comprising said file, if
15 any; and
- the length of valid data in said data portion of said memory block.
2. The file system for a nonvolatile memory medium according to claim 1,
wherein said header portion further comprises a flag indicating whether said file is
20 a secure file.

3. The file system for a nonvolatile memory medium according to claim 1, wherein said header portion further comprises the block number of the previous memory block, if any.

5 4. The file system for a nonvolatile memory medium according to claim 1, wherein said header portion further comprises the block number of the previous memory block, if any, and a flag indicating whether said file is a secure file.

5. A nonvolatile memory medium, characterized in that it is logically divided
10 into memory blocks of predetermined size, wherein each of said memory blocks comprises a header portion and a data portion, and wherein said header portion comprises:

a magic number indicating whether said memory block is a valid block or a free block;

15 the name of the file to which said memory block belongs;

a flag indicating whether said memory block is the first or last block of said file;

the block number of the next memory block comprising said file, if any;

and

20 the length of valid data in said data portion of said memory block.

6. The nonvolatile memory medium according to claim 5, wherein said header portion further comprises a flag indicating whether said file is a secure file.

7. The nonvolatile memory medium according to claim 5, wherein said header portion further comprises the block number of the previous memory block, if any.

8. The nonvolatile memory medium according to claim 5, wherein said header portion further comprises the block number of the previous memory block, if any, and a flag indicating whether said file is a secure file.

10

9. A method for opening an existing file stored on a nonvolatile memory medium, comprising:

(1) scanning said nonvolatile memory medium to find a first memory block containing a header indicating that said first memory block is the first memory block of said file; and

(2) finding a next memory block using a next block pointer stored in said header of said first memory block, if said file comprises more than said first memory block.

10. The method of claim 10, further comprising the step of repeating step (2) until either all memory blocks comprising said file have been found or an error condition occurs.

11. A method for opening an existing file stored on a nonvolatile memory medium, comprising:

scanning said nonvolatile memory medium in sizes of one predetermined
5 logical block;

for each said logical block, reading a block header containing a magic number;

testing said magic number to determine whether said logical block is a valid block or a free block, and if said logical block is a valid block, performing a
10 comparison of a file name encoded within said block header with a specified file name to be opened;

testing a flag within said block header indicating to determine whether said logical block is the first block of said specified file, if said comparison produces a match; and

15 returning to said scanning step with the next logical block until either said comparison produces a match or all the blocks have been tested, thereby indicating an error condition.

12. A method for opening a new file to be stored on a nonvolatile memory
20 medium, comprising:

scanning said nonvolatile memory medium in sizes of one predetermined logical block;

for each said logical block, reading a block header containing a magic number;

testing said magic number to determine whether said logical block is a valid block or a free block, and if said logical block is a free block, modifying said
5 block header to comprise a valid magic number, the name of the file to be opened, and flags indicating whether said logical block is either the first block or the last block of said file; and

returning to said scanning step with the next logical block until either said testing step has identified a free block or all the blocks have been tested, thereby
10 indicating an error condition.

13. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method for opening an existing file stored on a nonvolatile memory medium, the method
15 comprising:

scanning said nonvolatile memory medium to find a first memory block containing a header indicating that said first memory block is the first memory block of said file; and

finding a next memory block using a next block pointer stored in said
20 header of said first memory block, if said file comprises more than said first memory block.

14. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method for opening an existing file stored on a nonvolatile memory medium, the method comprising:

5 scanning said nonvolatile memory medium in sizes of one predetermined logical block;

for each said logical block, reading a block header containing a magic number;

testing said magic number to determine whether said logical block is a valid
10 block or a free block, and if said logical block is a valid block, performing a comparison of a file name encoded within said block header with a specified file name to be opened;

testing a flag within said block header indicating to determine whether said logical block is the first block of said specified file, if said comparison produces a
15 match; and

returning to said scanning step with the next logical block until either said comparison produces a match or all the blocks have been tested, thereby indicating an error condition.

20 15. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method for opening a new file to be stored on a nonvolatile memory medium, comprising:

scanning said nonvolatile memory medium in sizes of one predetermined logical block;

for each said logical block, reading a block header containing a magic number;

5 testing said magic number to determine whether said logical block is a valid block or a free block, and if said logical block is a free block, modifying said block header to comprise a valid magic number, the name of the file to be opened, and flags indicating whether said logical block is either the first block or the last block of said file; and

10 returning to said scanning step with the next logical block until either said testing step has identified a free block or all the blocks have been tested, thereby indicating an error condition.

16. An apparatus for implementing a file system having a flat file structure,
15 comprising:

 a processor configured to control the writing of data to and reading of data from a nonvolatile memory medium, wherein said nonvolatile memory medium is logically divided into memory blocks of predetermined size, wherein each of said memory blocks comprises a header portion and a data portion, and wherein said
20 header portion comprises a magic number indicating whether said memory block is a valid file system block, the name of the file to which said memory block belongs, if any, a set of flags indicating whether said memory block is the first or

last block of said file, the block number of the next memory block comprising said file, if any, and the length of valid data in said data portion of said memory block.

17. The apparatus according to claim 16, wherein said header portion further
5 comprises a flag indicating whether said file is a secure file.

18. The apparatus according to claim 16, wherein said header portion further comprises the block number of the previous memory block, if any.

10 19. The apparatus according to claim 16, wherein said header portion further comprises the block number of the previous memory block, if any, and a flag indicating whether said file is a secure file.

15